An Optimized LSTM-Based Strategy for Exercise Type Recognition Using Fitness Tracker Data

Samad Najjar-Ghabel[a], Shamim Yousefi*[a], Bakr Ammar Jabbar[a]

[a]*Department of Computer Engineering, University of Mohaghegh Ardabili, Ardabil, Iran*

Abstract: Automatic recognition of exercise types using data from wearable sensors is a key challenge in the field of digital health and fitness. This study presents a machine learning–based strategy for classifying physical activities using data collected from fitness trackers. We employ a cost-sensitive and optimized Long Short-Term Memory (LSTM) network to better handle class imbalances and model temporal dependencies in the sensor data. During preprocessing, raw signals are cleaned, normalized, and transformed into meaningful features. The customized LSTM model is then trained to learn hidden patterns with an emphasis on minimizing misclassification costs. Evaluation using standard performance metrics shows that the proposed cost-sensitive LSTM approach significantly outperforms traditional methods, achieving an accuracy of 99%. Precision, recall, and F1-score also indicate excellent performance, demonstrating the method's strong capability in accurate activity recognition. Unlike many previous studies focused on non-sequential or standard LSTM models, this research highlights the advantages of tailored recurrent architectures in enhancing the robustness and accuracy of exercise classification systems.

Keywords:

Deep learning
Recurrent neural networks
LSTM
Sports activity classification
Fitness tracker

1. Introduction

The rapid proliferation of wearable fitness devices has revolutionized the tracking of physical activities, enabling the collection of diverse physiological and motion data such as accelerometer and gyroscope signals [1, 2]. These devices hold immense potential for automated exercise recognition, a critical task in digital health and personalized fitness monitoring. However, accurately classifying exercise types from sensor data remains challenging due to inherent complexities such as sensor noise, inter-activity similarity, temporal dependencies in motion patterns, and data imbalance across exercise categories [3, 4, 5, 6]. Traditional machine learning approaches, which often rely on handcrafted features and non-sequential models, struggle to capture the dynamic and context-dependent nature of human movements, leading to suboptimal performance in real-world scenarios [7, 8, 9].

Recent advancements in deep learning, particularly recurrent neural networks (RNNs), offer promising solutions for temporal data analysis. Long Short-Term Memory (LSTM) networks, a specialized variant of RNNs, excel at modeling long-range dependencies in time-series data, making them well-suited for sensor-based activity recognition [10, 11]. While prior studies have explored classical machine learning methods and basic neural architectures, their focus on static or non-sequential features often neglects the rich temporal correlations inherent in exercise movements. Furthermore, existing approaches frequently suffer from limited generalizability across diverse user populations and varying sensor configurations, highlighting the need for robust, adaptive frameworks [12].

This study introduces a novel LSTM-based strategy to address these limitations, emphasizing the extraction of temporal patterns from raw sensor data for precise exercise classification. Unlike conventional methods that rely on manual feature engineering, our approach leverages the inherent capability of LSTMs to autonomously learn discriminative spatiotemporal features from accelerometer and gyroscope signals.

Key innovations include:

- A sliding window technique optimized for capturing motion dynamics.
- A hybrid preprocessing pipeline integrating moving average filters and Z-score normalization to mitigate noise and sensor variability.

**Corresponding author.*
E-mail addresses: Sh.yousefi@uma.ac.ir

- A compact LSTM architecture tailored for resource-efficient deployment on wearable devices.

Experimental validation on a comprehensive fitness tracker dataset demonstrates the superiority of our framework, achieving 99% accuracy in classifying six distinct exercise types, significantly outperforming traditional classifiers [13]. The model also exhibits robust performance metrics, with precision, recall, and F1-scores, underscoring its effectiveness in real-world applications. These results not only validate LSTM's efficacy for temporal exercise recognition but also provide insights into optimizing computational efficiency for edge deployment.

This work advances the field by bridging the gap between high accuracy and practicality in wearable-based activity recognition. It establishes LSTM networks as a superior alternative to classical methods while addressing critical challenges such as noise resilience and temporal feature extraction.

2. Related Work

Traditional human activity recognition (HAR) systems using wearable sensors have historically depended on handcrafted features paired with shallow classifiers such as decision trees, support vector machines (SVM) and k-nearest neighbors (k-NN) [14, 15]. Random forests and Gaussian mixture models (GMMs) have also been popular for their robustness and interpretability [16]. Although these methods perform well on clean, well-segmented datasets, they require extensive feature engineering to handle user variability and real-world noise. The advent of deep learning, especially convolutional neural networks (CNNs) and recurrent neural networks (RNNs), has transformed HAR. CNNs excel at extracting local spatiotemporal patterns from inertial data, while RNNs, notably long short-term memory (LSTM) networks, capture longer-term temporal dependencies in sequential signals [13, 17]. Hybrid architectures such as DeepConvLSTM, which stack CNN and LSTM layers, have delivered state-of-the-art accuracy in recognizing complex motion directly from raw sensor streams [17].

Effective preprocessing remains crucial for mitigating sensor noise and enhancing feature relevance. Techniques such as sliding-window segmentation, Z-score normalization, moving-average filtering and time- and frequency-domain feature extraction improve model performance [18, 19]. However, manual preprocessing still limits adaptability and scalability [20]. Wearable data are prone to motion artifacts and device misplacement. Denoising strategies and temporal modeling with LSTMs can alleviate these issues, while autoencoder-based approaches learn robust features directly from noisy raw data [20, 21]. Generalization across users is a persistent challenge because subject-independent evaluations, for example leave-one-subject-out, often yield lower accuracy than random splits. This underscores the need for models that handle inter-user variability. Solutions include normalization, data augmentation and transfer learning to improve cross-subject performance [22, 20].

Real-time, on-device HAR on resource-constrained hardware requires lightweight models. Model pruning, quantization and compact CNN–LSTM designs enable efficient inference with minimal accuracy loss [23, 24]. For example, Merenda et al. [25] implemented exercise detection and repetition counting directly on edge devices using CNN models with handcrafted features, achieving 100% accuracy in real-time without cloud connectivity. Similarly, Jamil et al. [26] proposed a secure fitness framework integrating IoT-enabled blockchain with a statistical ML inference engine, achieving 92% accuracy without sequential modeling. Vision-based HAR is also gaining traction; Cholaraja et al. [27] employed CNN-based computer vision techniques for real-time exercise monitoring, repetition counting and posture analysis, although their approach does not use feature engineering or sequential modeling. Veeraiah et al. [28] applied SVM classifiers with handcrafted features to analyze user activity logs from fitness trackers, focusing on calorie expenditure and behavioral insights rather than real-time classification.

These studies illustrate the diversity of methodologies in exercise type recognition, varying in their use of feature engineering, sequential modeling, classifier types and application focus. Our work builds on these foundations by introducing an optimized cost-sensitive LSTM model that combines temporal sequence learning with class imbalance handling, enabling more accurate and robust exercise recognition from wearable sensor data in realistic conditions.

3. Methodology

This study develops a machine learning pipeline to classify exercise types using inertial data collected from wrist-worn fitness trackers. The workflow consists of four key stages: data preprocessing, feature engineering, windowing, and model development with optimization. The overall framework is illustrated in Figure 1.

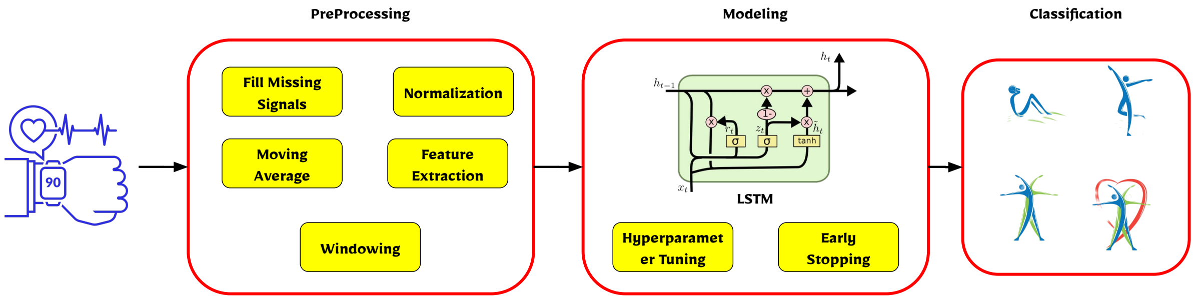

Figure 1: A Machine Learning–Based Strategy for Exercise Type Recognition Using Fitness Tracker Data.

3.1. Data Preprocessing

The raw accelerometer and gyroscope signals used in this study were obtained from consumer-grade fitness trackers. These signals are inherently subject to noise, missing values, and variable sampling rates. Such imperfections can significantly degrade the performance of downstream machine learning models. Therefore, a rigorous preprocessing pipeline was implemented to ensure data quality and consistency before feature extraction and model training.

The first step in preprocessing involved addressing noise and artifacts in the raw signal. Missing and invalid values were identified by inspecting the time series for gaps or undefined entries. Short gaps, defined as fewer than five consecutive samples, were imputed using linear interpolation. In contrast, longer gaps were flagged and excluded from further analysis to avoid introducing bias or unreliable patterns. To manage outliers and sudden jumps, rolling z-scores were computed with a window of 50 samples on each axis of the sensor data. Any segment with extreme deviations, defined by $|z| > 3$, was treated as an outlier. These segments were corrected using local median filtering to maintain signal continuity without distorting the underlying movement patterns.

Next, a noise reduction step was performed using a centered moving average filter. This technique was applied independently to each axis of both the accelerometer and gyroscope signals. The window length for filtering was chosen adaptively based on the nominal sampling rate of each recording, typically corresponding to a one-second interval. This approach ensured effective suppression of high-frequency noise while preserving essential temporal features related to exercise movements. Figure 2 illustrates the effect of this filtering process on a representative accelerometer signal. The noise-filtering methods employed in this study, such as moving average filtering, were manually tuned for the current dataset to mitigate common sensor noise and enhance feature quality. While these techniques are effective for the conditions considered, we acknowledge that their performance may vary across different devices or in noisier environments.

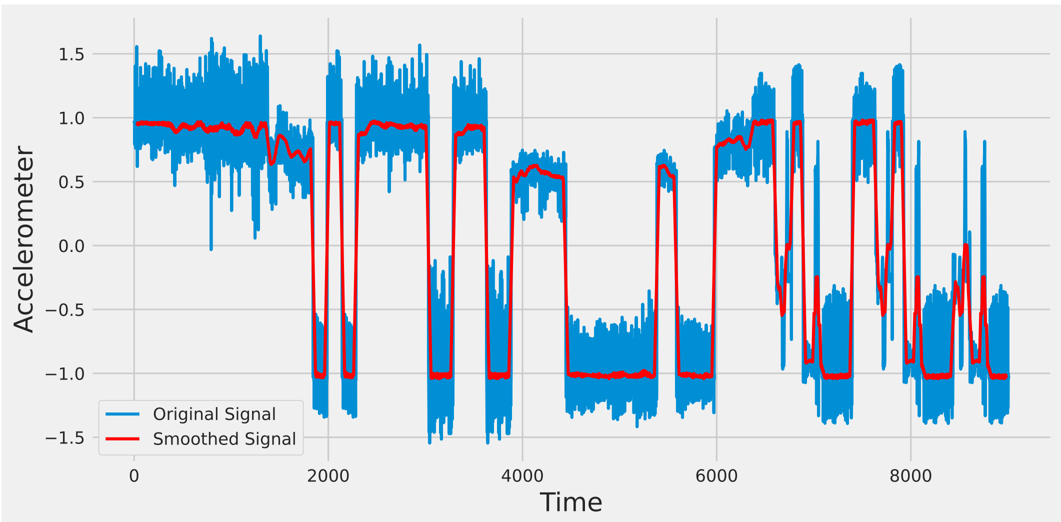

Figure 2: Output of the moving average filter on a representative accelerometer signal.

The final stage of preprocessing involved normalization through Z-score standardization. This step was necessary because sensor channels often operate over different dynamic ranges—for example, $\pm 16\,g$ for acceleration versus

$\pm 2000\,°/s$ for angular velocity. Without normalization, features with larger magnitudes could disproportionately influence gradient-based learning algorithms. For each sensor axis x, the standardized signal $\tilde{x}$ was computed as:

$$\tilde{x} = \frac{x - \mu_x}{\sigma_x},$$

where μ_x and σ_x represent the mean and standard deviation of x, respectively, calculated over the entire recording. This transformation ensured that each sensor channel had zero mean and unit variance. In addition to improving numerical stability, standardization enhanced the convergence speed of optimization algorithms and improved the generalizability of the resulting models.

3.2. Feature Engineering

After the preprocessing of raw data, the feature engineering stage was conducted to extract informative characteristics from the sensor signals. In this stage, features were designed and computed to uncover latent patterns within the data that could enhance the predictive power of the machine learning models. The extracted features fall into three main categories: statistical, frequency-based, and temporal features, each contributing uniquely to the characterization of physical activities.

A particularly important derived feature is the signal magnitude, computed from the accelerometer and gyroscope data. For a given three-axis signal with components x, y, and z, the magnitude m is calculated as:

$$m = \sqrt{x^2 + y^2 + z^2}.$$

This magnitude feature offers several advantages. First, variations in signal magnitude often correlate with the intensity and type of physical activity. For example, high-intensity movements such as running or jumping typically result in larger magnitude values. Second, because magnitude is independent of the device's orientation, this feature helps reduce model sensitivity to the position and orientation of the wrist-worn tracker, thereby improving the robustness and generalizability of the classification system.

In addition to computing the magnitude itself, several time-domain statistical descriptors of the magnitude signal were also calculated, including the mean, standard deviation, and range over fixed-length sliding windows. These descriptors provide insights into the temporal evolution and variability of movement patterns, further enhancing the model's ability to differentiate among activity classes.

Together, these engineered features establish a rich, multidimensional representation of user movement, laying the groundwork for accurate and interpretable activity classification in the subsequent model development phase.

3.3. Windowing

To effectively process continuous-time signals for use in recurrent models, a moving window technique was employed. This approach divides the time series data into smaller, sequential segments, enabling the model to better capture temporal dependencies and identify patterns and transitions over time. A fixed-length window slides over the data, extracting a new segment at each step. This preserves temporal structure and allows the model to detect subtle and evolving patterns more effectively.

An important parameter in this process is the moving step size, which determines the offset between consecutive windows. A small step size results in high overlap between successive windows, preserving more detail and potentially enhancing model performance. However, this comes at the cost of increased input volume and higher computational demands. Conversely, a larger step size reduces computational load by lowering overlap, but may sacrifice critical information and reduce accuracy. Hence, selecting an appropriate step size involves balancing granularity and computational efficiency.

Another critical factor is the window length, which defines the temporal span of each segment. A short window may fail to capture longer-term dependencies, thereby diminishing the model's ability to recognize complex activities. Conversely, excessively long windows might smooth over short-term variations, making it harder to distinguish between distinct activity types. In the context of human activity recognition from wearable sensor data, window lengths between 1 and 10 seconds are typically used. This range effectively captures both brief and extended activity patterns.

Overall, the windowing process supports the recurrent architecture by segmenting the data into informative, temporally coherent units. It enables recurrent models to learn long-term relationships, improves classification accuracy, and enhances the overall efficiency of the machine learning pipeline.

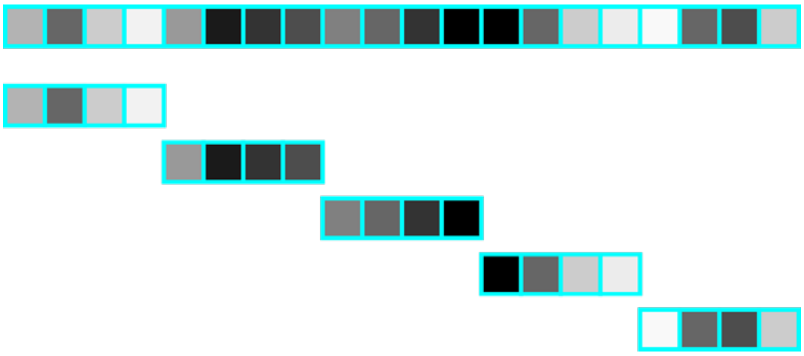

Figure 3: Illustration of the moving windowing technique used for segmenting time series data.

3.4. Modeling

In this study, a cost-sensitive Long Short-Term Memory (LSTM) network was used to process time series data and extract hidden patterns in sensor signals. LSTM is a type of recurrent neural network (RNN) capable of learning long-term dependencies through a specialized structure consisting of memory cells and gating mechanisms. Unlike conventional RNNs, which suffer from the vanishing gradient problem, LSTMs overcome this issue by controlling information flow through input, forget, and output gates. To address class imbalance and improve recognition of minority classes, the LSTM architecture was customized with a cost-sensitive learning mechanism, which penalizes misclassifications of underrepresented activities more heavily during training. This adaptation enables the model to learn more robust temporal features and enhances its ability to generalize across different exercise types [29].

The main components of the LSTM unit include:

1. **Forget Gate:** Determines which parts of the cell state should be discarded. This is essential for removing irrelevant or noisy information from memory [30].
2. **Input Gate:** Identifies which new information should be added to the cell state. It uses a combination of sigmoid and hyperbolic tangent functions to select relevant values [31].
3. **Output Gate:** Regulates which parts of the cell state are output to the next time step and updated as hidden states [32].

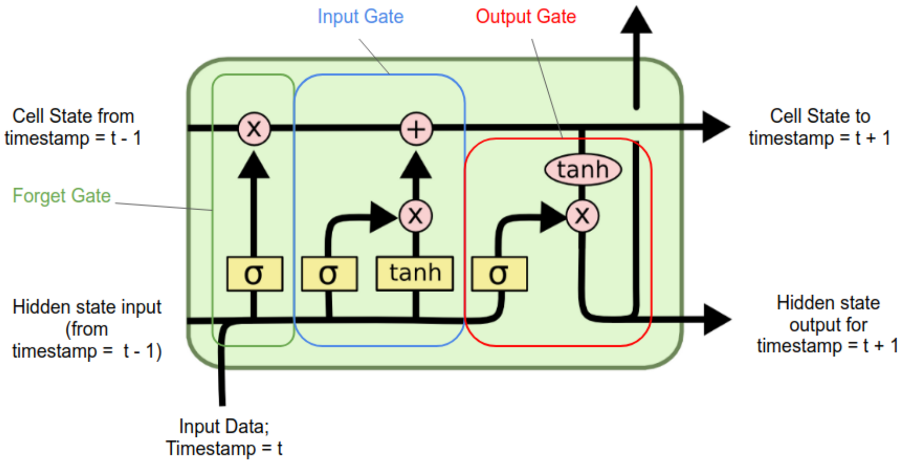

Figure 4: An LSTM unit [33].

The model developed for this study begins with an LSTM layer consisting of a configurable number of memory units to capture long-range dependencies in sequential data. This is followed by a dense (fully connected) layer with 32 neurons and a ReLU activation function, which helps extract non-linear relationships and improve model generalization. The final layer is a dense output layer with 6 neurons, each corresponding to one activity class, using a Softmax activation function to output class probabilities.

For training, the Adam optimizer was used due to its adaptive learning rate and proven effectiveness in deep learning tasks. The loss function employed was categorical cross-entropy, which is appropriate for multi-class classification problems. Model evaluation was conducted using accuracy as the primary metric, quantifying the proportion of correctly classified activity samples.

To address class imbalance in the dataset, class weights were incorporated into the loss function of the LSTM model. Each class was assigned a weight inversely proportional to its frequency in the training data. This cost-sensitive approach penalizes misclassification of minority classes more heavily, enabling the model to learn effectively from imbalanced data without resampling or altering the original dataset distribution.

3.5. Model Optimization

To enhance classification accuracy and ensure robust model generalization, a series of model optimization strategies were employed. These included hyperparameter tuning, network search, and validation techniques such as cross-validation and early stopping.

Hyperparameter tuning was performed using a grid search approach to systematically explore different model configurations. For each candidate configuration, parameters such as the number of LSTM layers, the number of hidden units per layer, learning rate, batch size, and dropout rate were varied within ranges informed by prior experiments and related literature.

The grid search was carried out in a subject-independent evaluation setting to ensure that the model could generalize to unseen users. Within each cross-validation fold, the dataset was split into training and validation subsets, and early stopping was applied to prevent overfitting. Validation performance, measured primarily by F1-score, was used to select the optimal set of parameters.

This process ensured that the final model configuration represented the best trade-off between accuracy, generalization capability, and training stability.

3.5.1. Hyperparameter Tuning

The model's performance was significantly influenced by key hyperparameters, including:

- Number of LSTM units
- Learning rate of the optimizer
- Window size and step (from the windowing phase)
- Number of neurons in the fully connected layer

A grid search approach was adopted to systematically explore combinations of these parameters. For each hyperparameter, a discrete set of possible values was defined. The model was trained with all parameter combinations, and the configuration yielding the best validation performance was selected.

3.5.2. Cross-Validation

To validate the model's ability to generalize across unseen data, k-fold cross-validation was applied with $k = 5$. The dataset was partitioned into five equally sized subsets, and the model was trained and validated five times, each time using a different subset as validation data and the remaining four subsets for training. This choice of k was made to provide a balanced trade-off between computational cost and the stability of performance estimates.

This method provided a reliable estimate of model performance and helped reduce overfitting by ensuring that every data instance was used for both training and validation across different folds.

3.5.3. Early Stopping

To avoid overfitting and reduce training time, early stopping was implemented. During training, if the model's performance on the validation set did not improve for 5 consecutive epochs (patience = 5), training was halted. This prevented the model from learning noise and ensured better generalizability.

Overall, the combination of grid search, 5-fold cross-validation, and early stopping significantly improved model performance. These optimization techniques led to higher classification accuracy, reduced generalization error, and more robust activity recognition across different users and sensor conditions.

4. Experimental Results

This section presents the results obtained from the analysis and modeling of accelerometer and gyroscope data collected from participants during various exercises. The main objective is to evaluate the performance of the developed machine learning models in detecting and classifying sports activities based on inertial sensor data. The chapter provides a comprehensive description of the dataset, quantitative and qualitative assessments of the model performance, comparisons between different modeling approaches, and a discussion of the key findings.

Initially, the dataset used in this research and its characteristics are fully described. Subsequently, the numerical results of the models are presented. Several machine learning and deep learning models were selected and trained to detect sports activities; their performances are analyzed from multiple perspectives. Additionally, the training process is examined by visualizing the convergence trends and the evolution of training and validation errors to assess model stability.

A comparative analysis of the different models follows, highlighting which approaches yield superior performance in classifying sports activities based on the selected evaluation criteria. Finally, the chapter summarizes the key insights derived from the experimental results.

4.1. Dataset Description

The dataset employed in this study, referred to as the Fitness Tracker dataset, comprises accelerometer and gyroscope sensor data collected from participants performing various exercises. The sensor readings include three-axis accelerometer (x, y, z) and gyroscope (x, y, z) data across multiple exercise categories and intensities. This dataset is well-suited for analyzing movement patterns, developing activity recognition models, and training machine learning algorithms in fitness and health monitoring applications. The dataset consists of 9009 samples, each containing 11 features, including 6 sensor-based measurements and 5 metadata fields.

- **Main Features of the Dataset:**
  - **Epoch (ms):** Timestamp in milliseconds indicating the exact moment of data recording.
  - **Accelerometer_x/y/z:** Accelerometer readings along the x, y, and z axes.
  - **Gyroscope_x/y/z:** Gyroscope rotational speed measurements along the x, y, and z axes.
  - **Participants:** Identifier of the individual performing the exercise.
  - **Label:** Type of exercise or movement performed (e.g., bench press, overhead press).
  - **Category:** Exercise intensity classification (e.g., heavy, moderate).
  - **Set:** Set number or session identifier during data collection.

The dataset spans recordings collected over 10 days and offers detailed sensor information across various activities, facilitating in-depth analysis and the development of robust human activity recognition models. Figure 5 illustrates the distribution of different exercise types in the dataset, demonstrating a near-even class distribution.

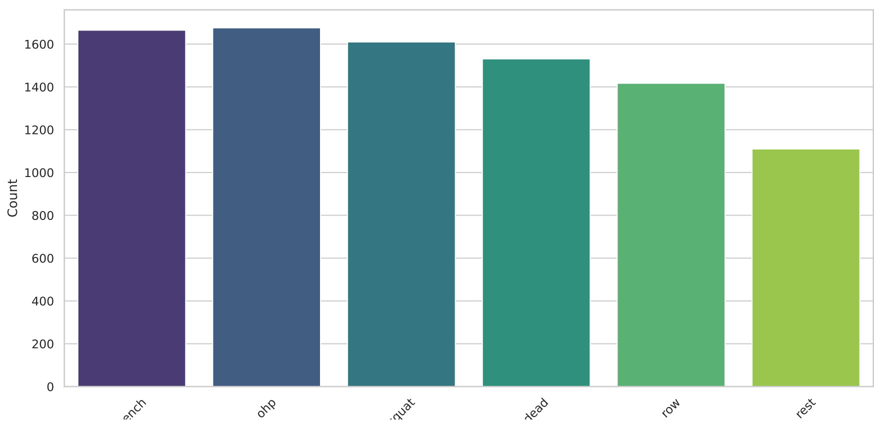

Figure 5: Distribution of different exercises across the dataset.

Figure 6 presents a three-dimensional visualization of the dataset, showing that while some exercise classes are well separated, others exhibit overlap, indicating the challenge of discriminating between certain activities based on sensor data.

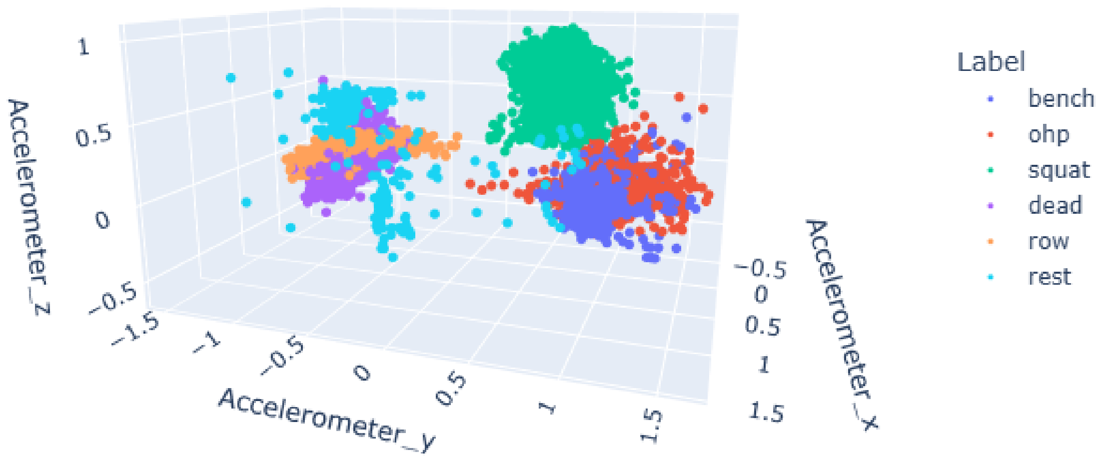

Figure 6: Three-dimensional distribution of the dataset across selected features.

Figure 7 shows the correlation matrix of the dataset features. The matrix reveals no pairs of features with extremely high linear correlation, supporting the inclusion of all features for model training without significant redundancy.

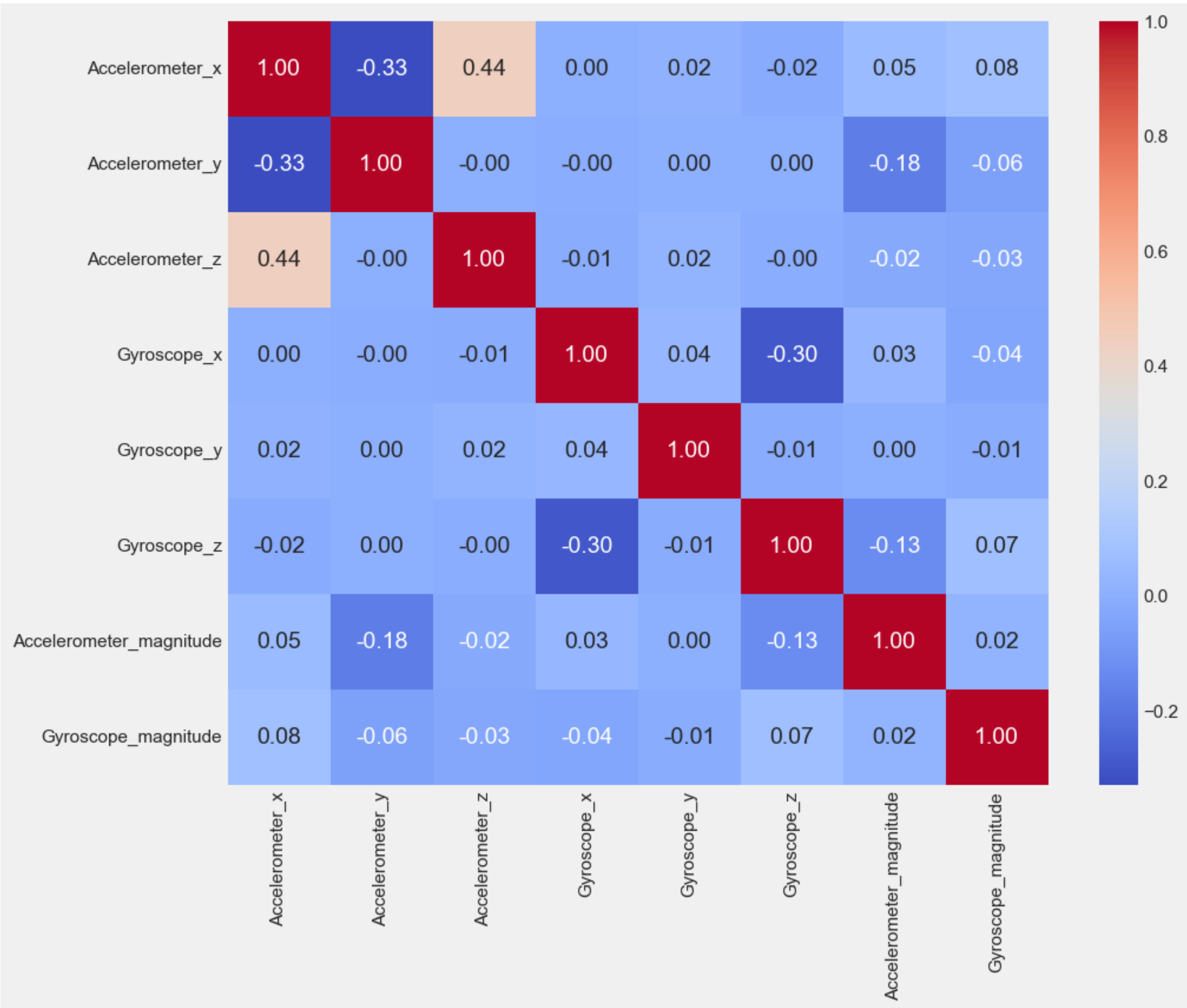

Figure 7: Correlation matrix of the dataset features.

4.2. Hardware and Execution Time

All training and inference experiments were performed on a system with the following specifications: Intel Core i7-1355U processor (1.7–5.0 GHz), 16 GB DDR4 RAM with a frequency of 3200 MHz.

Execution times reported in this study were measured on this configuration to ensure that the comparisons are meaningful and reproducible. All models were run using the same environment and hardware to provide a consistent basis for performance evaluation.

4.3. Results of Modeling and Data Analysis

In this section, the results of modeling and data analysis are evaluated. First, the dataset was modeled and evaluated using classical machine learning methods. Table 1 presents the comparison of various classical classifiers based on metrics such as accuracy, balanced accuracy, F1 score, and processing time.

Table 1: Comparison of Classical Methods

Model	**Accuracy**	**Balanced Accuracy**	**F1 Score**	**Time (s)**
ExtraTreeClassifier	0.84	0.87	0.89	0.04
LGBMClassifier	0.83	0.86	0.88	0.81
KNeighborsClassifier	0.83	0.85	0.89	0.33
ExtraTreesClassifier	0.81	0.84	0.87	0.50
LabelPropagation	0.81	0.84	0.87	2.35
LabelSpreading	0.81	0.84	0.87	3.46
BaggingClassifier	0.80	0.83	0.86	1.03
RandomForestClassifier	0.80	0.83	0.86	1.87
DecisionTreeClassifier	0.79	0.82	0.85	0.15
Quadratic Discriminant Analysis	0.75	0.78	0.81	0.04
SVC	0.63	0.68	0.64	0.73
NearestCentroid	0.62	0.68	0.69	0.04
SGDClassifier	0.61	0.67	0.62	0.11
LogisticRegression	0.60	0.66	0.61	0.16
GaussianNB	0.62	0.66	0.63	0.04
LinearSVC	0.60	0.66	0.59	0.07
Linear Discriminant Analysis	0.60	0.65	0.61	0.06
CalibratedClassifierCV	0.59	0.65	0.59	0.52
PassiveAggressiveClassifier	0.58	0.63	0.58	0.08
BernoulliNB	0.56	0.62	0.58	0.27
AdaBoostClassifier	0.55	0.62	0.60	1.72
Perceptron	0.57	0.60	0.58	0.06
RidgeClassifier	0.52	0.59	0.55	0.04
RidgeClassifierCV	0.52	0.59	0.55	0.08
DummyClassifier	0.00	0.00	0.00	0.04

A review of the results in Table 1 indicates that the **ExtraTreeClassifier** and **LGBMClassifier** achieved the best performance in terms of accuracy, balanced accuracy, and F1 score. Among these, ExtraTreeClassifier was more efficient due to its lower processing time. Models such as *RandomForestClassifier* and *BaggingClassifier* delivered moderate performance, while requiring higher computation time. In contrast, linear models including *LogisticRegression*, *GaussianNB*, and *LinearSVC* performed poorly and appear unsuitable for this dataset. The *DummyClassifier* and simpler models like *RidgeClassifier* and *BernoulliNB* showed the weakest performance. Therefore, decision tree-based models like ExtraTreeClassifier and LGBMClassifier stand out as optimal choices for this data due to their accuracy and efficiency. Next, we evaluated the performance of recurrent neural network models. Table 2 presents the results of various recursive models, including Simple RNN, GRU, and our proposed LSTM model.

Table 2: Comparison of Recurrent Methods

Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**Time (s)**
Simple RNN	0.95	0.95	0.95	0.95	50
GRU	0.98	0.98	0.98	0.98	55
LSTM (Proposed)	**0.99**	**0.99**	**0.99**	**0.99**	58

According to Table 1 and 2, the proposed LSTM-based model outperforms all classical and recurrent methods, achieving the highest accuracy, F1 score, and recall. While it incurs a slightly higher processing time, the trade-off is justified by its superior classification performance. As shown in Figure 8, the confusion matrix of the proposed LSTM model confirms its high effectiveness in accurately detecting all classes with minimal misclassifications.

Notably, each activity class is predicted with high confidence and precision. The diagonal dominance of the matrix highlights the model's robust classification ability, with the majority of samples for each exercise type, such as bench press, deadlift, overhead press (ohp), and squat, being correctly identified. Misclassifications are rare and minor, often involving confusion between exercises with similar motion patterns, such as row and ohp. For instance, only a few instances of "row" were misclassified as "ohp", reflecting the model's nuanced understanding of overlapping activity features. These minimal errors further emphasize the model's discriminative power and its capability to generalize across diverse exercise movements. Overall, the LSTM model demonstrates strong potential for deployment in real-time fitness tracking applications due to its balance between accuracy and computational efficiency.

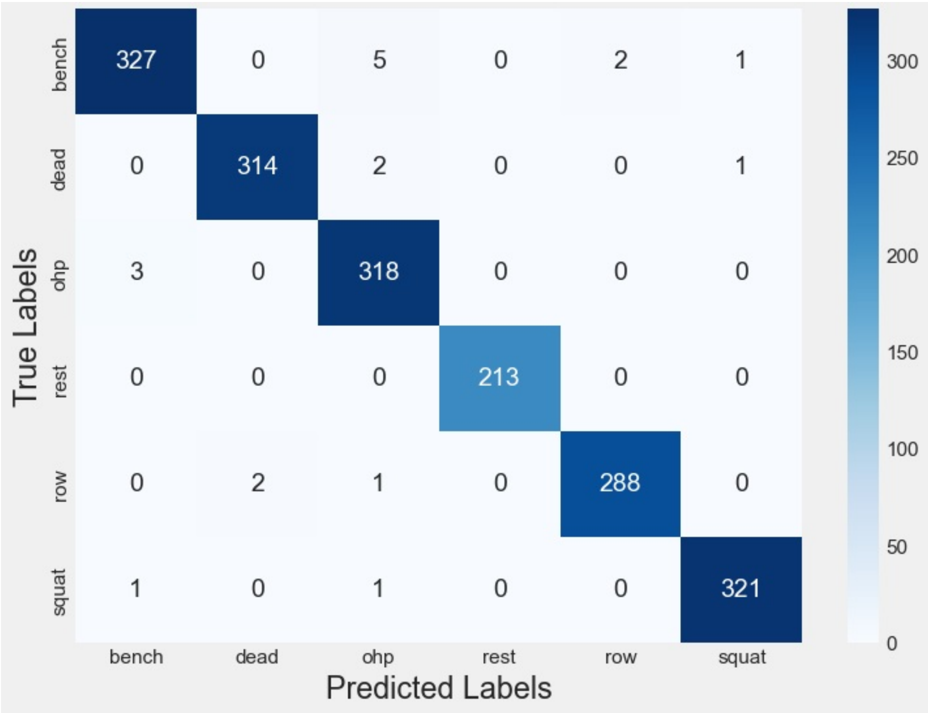

Figure 8: Confusion Matrix of the Proposed LSTM Model

4.4. Impact of Noise on Classification Performance

To evaluate the robustness of the proposed model, we analyzed the effect of noise on both features and labels. Figure 9 shows the F1-score of the model under varying levels of noise. Feature noise was introduced by adding Gaussian noise at different percentages, while label noise was simulated by randomly flipping a percentage of labels.

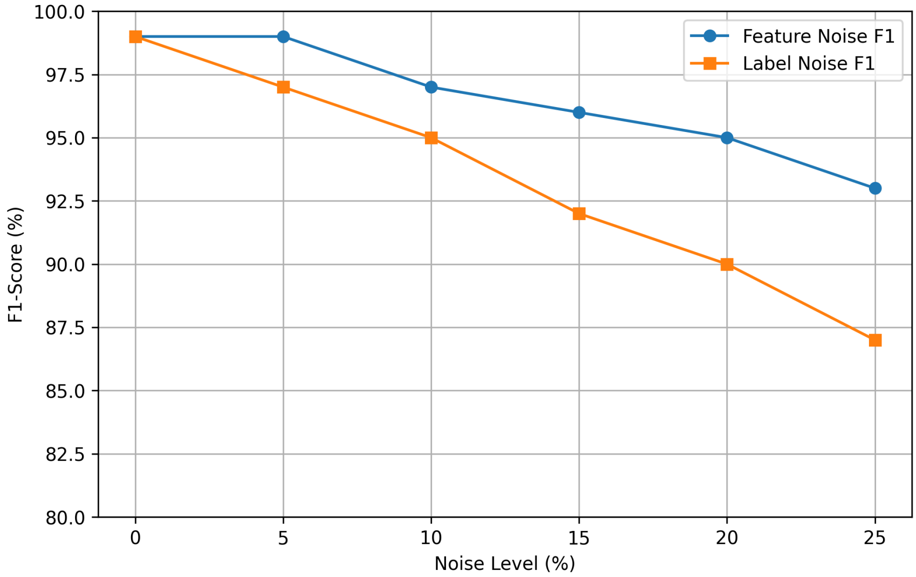

Figure 9: Effect of varying levels of noise in features and labels on F1-score.

The results demonstrate that while performance decreases as noise increases, the proposed cost-sensitive LSTM model maintains reasonable accuracy under moderate noise levels, illustrating its robustness.

5. Conclusion and Future Work

In this study, a machine learning-based strategy was proposed for recognizing types of physical exercises using data collected from fitness trackers. The dataset included accelerometer and gyroscope signals that capture detailed information about body movements during various exercise activities. Unlike traditional approaches that largely rely on classical machine learning models and non-sequential techniques, this study focused on leveraging temporal dependencies through the use of cost-sensitive Long Short-Term Memory (LSTM) networks. The proposed cost-sensitive LSTM model demonstrated a strong capability in extracting hidden features and learning complex temporal patterns from sequential sensor data, which significantly improved the classification performance. One of the key challenges addressed was the presence of noise and abrupt signal variations, which were mitigated using preprocessing techniques such as moving average filtering and data normalization. The evaluation results revealed that the LSTM-based approach achieved superior performance across all major evaluation metrics, including accuracy, precision, recall, and F1 score. Comparative analysis with conventional classifiers such as Decision Trees, Random Forests, and Support Vector Machines indicated that the LSTM model provided a marked improvement in activity recognition, particularly due to its ability to model temporal relationships effectively. Overall, the findings support the conclusion that recurrent neural networks, and LSTM in particular, offer a powerful and efficient solution for activity recognition tasks based on wearable sensor data. This approach has promising applications in developing intelligent exercise monitoring systems and enhancing personalized fitness programs.

While the proposed model achieved high accuracy, there is still room for improvement. One notable limitation is the relatively high computational cost and processing time associated with LSTM networks, which may hinder their use in real-time applications. Future work could focus on optimizing the model architecture or exploring more efficient alternatives, such as bidirectional or attention-based recurrent models, to reduce inference time while maintaining accuracy.

Another potential direction is the integration of multimodal sensor data to provide a more holistic understanding of physical activities and improve the model's ability to distinguish between similar movements. Additionally, adopting advanced training techniques such as transfer learning or self-supervised learning could be beneficial, particularly in scenarios where labeled training data is scarce. These methods can enable the model to generalize better across different user profiles and environments, increasing the robustness and scalability of the system. Ultimately, future efforts should aim to enhance the efficiency, adaptability, and real-world applicability of the system for broader deployment in health monitoring and smart fitness technologies.

References

[1] J. Yang, M. Nguyen, P. P. San, X. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," *IJCAI*, 2015.

[2] L. Piwek, D. A. Ellis, S. Andrews, and A. Joinson, "The rise of consumer health wearables: Promises and barriers," *PLOS Medicine*, vol. 13, no. 2, p. e1001953, 2016.

[3] N. Y. Hammerla, S. Halloran, and T. Ploetz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," *IJCAI*, 2016.

[4] M. Zeng, L. T. Nguyen, B. Yu, O. J. Mengshoel, J. Zhu, P. Wu, and J. Zhang, "Convolutional neural networks for human activity recognition using mobile sensors," in *6th International Conference on Mobile Computing, Applications and Services*, pp. 197–205, 2014.

[5] S. Yousefi and S. Najjar-Ghabel, "Automatic lung diseases identification using discrete cosine transform-based features in radiography images," *Journal of Information and Communication Technology*, vol. 55, no. 55, p. 124, 2023.

[6] S. Yousefi, S. Najjar-Ghabel, and H. Shafaei, "A technical analysis and practical implementation of machine learning algorithms for predicting survival in breast cancer patients," in *2024 9th International Conference on Computer Science and Engineering (UBMK)*, pp. 1168–1173, IEEE, 2024.

[7] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1192–1209, 2013.

[8] S. Najjar-Ghabel, S. Yousefi, and P. Habibi, "Comparative analysis and practical implementation of machine learning algorithms for phishing website detection," in *2024 9th International Conference on Computer Science and Engineering (UBMK)*, pp. 1–6, IEEE, 2024.

[9] S. Yousefi, S. Najjar-Ghabel, R. Danehchin, S. S. Band, C.-C. Hsu, and A. Mosavi, "Automatic melanoma detection using discrete cosine transform features and metadata on dermoscopic images," *Journal of King Saud University-Computer and Information Sciences*, vol. 36, no. 2, p. 101944, 2024.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] B. Yousefimehr and M. Ghatee, "A distribution-preserving method for resampling combined with lightgbm-lstm for sequence-wise fraud detection in credit card transactions," *Expert Systems with Applications*, vol. 262, p. 125661, 2025.

[12] J. Wang, Y. Chen, S. Hao, X. Peng, L. Hu, and H. Zhang, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognition Letters*, vol. 119, pp. 3–11, 2019.

[13] C. Ronao and S.-B. Cho, "Human activity recognition with smartphone sensors using deep learning neural networks," *Expert Systems with Applications*, vol. 59, pp. 235–244, 2016.

[14] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *SIGKDD Explorations*, vol. 12, no. 2, pp. 74–82, 2011.

[15] J. Yang, "Toward physical activity diary: motion recognition using simple acceleration features with mobile phones," in *Proceedings of the 1st international workshop on Interactive multimedia for consumer electronics*, pp. 1–6, 2009.

[16] F. Attal, S. Mohammed, M. Dedabrishvili, F. Chamroukhi, L. Oukhellou, and Y. Amirat, "Physical human activity recognition using wearable sensors," *Sensors*, vol. 15, no. 12, pp. 31314–31338, 2015.

[17] F. J. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.

[18] O. Baños, J.-M. Galvez, M. Damas, H. Pomares, and I. Rojas, "Window size impact in human activity recognition," *Sensors*, vol. 14, no. 4, pp. 6474–6499, 2014.

[19] H. F. Nweke, Y.-S. Teh, G. B. Mujtaba, and M. A. Al-Garadi, "Data fusion and multiple classifier systems for human activity detection and health monitoring: Review and open research directions," *Information Fusion*, vol. 46, pp. 147–164, 2019.

[20] Q. Ni, Z. Fan, L. Zhang, C. D. Nugent, I. Cleland, Y. Zhang, and N. Zhou, "Leveraging wearable sensors for human daily activity recognition with stacked denoising autoencoders," *Sensors*, vol. 20, no. 18, p. 5114, 2020.

[21] M. M. Hassan, Z. M. Uddin, A. Mohamed, and A. Almogren, "A robust human activity recognition system using smartphone sensors and deep learning," *Future Generation Computer Systems*, vol. 81, pp. 307–313, 2018.

[22] J. Zhu, R. San-Segundo, and J. M. Pardo, "Feature extraction for robust physical activity recognition," *Human-centric Computing and Information Sciences*, vol. 7, no. 1, pp. 1–16, 2017.

[23] P. Agarwal and M. Alam, "A lightweight deep learning model for human activity recognition on edge devices," *Procedia Computer Science*, vol. 167, pp. 2364–2373, 2020.

[24] T. Zebin, P. J. Scully, N. Peek, A. J. Casson, and K. B. Ozanyan, "Design and implementation of a convolutional neural network on an edge computing smartphone for human activity recognition," *IEEE Access*, vol. 7, pp. 38513–38520, 2019.

[25] M. Merenda, M. Astrologo, D. Laurendi, V. Romeo, and F. G. Della Corte, "A novel fitness tracker using edge machine learning," in *2020 IEEE 20th Mediterranean Electrotechnical Conference (MELECON)*, pp. 212–215, IEEE, 2020.

[26] F. Jamil, H. K. Kahng, S. Kim, and D.-H. Kim, "Towards secure fitness framework based on iot-enabled blockchain network integrated with machine learning algorithms," *Sensors*, vol. 21, no. 5, p. 1640, 2021.

[27] K. Cholaraja, M. Gururaj, V. Swasthika, and S. Gnanasambandam, "Ai based automated gym trainer using machine learning," in *2024 10th International Conference on Advanced Computing and Communication Systems (ICACCS)*, vol. 1, pp. 2046–2049, IEEE, 2024.

[28] V. Veeraiah, J. V. N. Ramesh, A. Koujalagi, V. Talukdar, A. Namdev, and A. Gupta, "Health fitness tracker system using machine learning based on data analytics," in *International Conference on Mobile Radio Communications & 5G Networks*, pp. 765–775, Springer, 2023.

[29] R. C. Staudemeyer and E. R. Morris, "Understanding lstm–a tutorial into long short-term memory recurrent neural networks," *arXiv preprint arXiv:1909.09586*, 2019.

[30] Z. Zhang, Z. Lv, C. Gan, and Q. Zhu, "Human action recognition using convolutional lstm and fully-connected lstm with different attentions," *Neurocomputing*, vol. 410, pp. 304–316, 2020.

[31] Y. Hua, Z. Zhao, R. Li, X. Chen, Z. Liu, and H. Zhang, "Deep learning with long short-term memory for time series prediction," *IEEE Communications Magazine*, vol. 57, no. 6, pp. 114–119, 2019.

[32] J. Zhang, Y. Zhu, X. Zhang, M. Ye, and J. Yang, "Developing a long short-term memory (lstm) based model for predicting water table depth in agricultural areas," *Journal of hydrology*, vol. 561, pp. 918–929, 2018.

[33] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: Lstm cells and network architectures," *Neural computation*, vol. 31, no. 7, pp. 1235–1270, 2019.